# **Ultra Professional Website Generator - Complete Setup Guide**

Een complete toolkit voor het bouwen van professionele websites met moderne best practices, VS Code + Git + Netlify workflow en geautomatiseerde kwaliteitscontrole.

## **Inhoudsopgave**

1. [Wat is dit?](https://claude.ai/chat/9e5a005f-cc9c-44d4-a437-9847e6da0821#wat-is-dit)
2. [Quick Start (5 minuten)](https://claude.ai/chat/9e5a005f-cc9c-44d4-a437-9847e6da0821#quick-start)
3. [Volledige Installatie](https://claude.ai/chat/9e5a005f-cc9c-44d4-a437-9847e6da0821#volledige-installatie)
4. [Project Structuur](https://claude.ai/chat/9e5a005f-cc9c-44d4-a437-9847e6da0821#project-structuur)
5. [Alle Bestanden Uitgelegd](https://claude.ai/chat/9e5a005f-cc9c-44d4-a437-9847e6da0821#alle-bestanden-uitgelegd)
6. [Development Workflow](https://claude.ai/chat/9e5a005f-cc9c-44d4-a437-9847e6da0821#development-workflow)
7. [Kwaliteitscontrole & Testing](https://claude.ai/chat/9e5a005f-cc9c-44d4-a437-9847e6da0821#kwaliteitscontrole--testing)
8. [Deployment](https://claude.ai/chat/9e5a005f-cc9c-44d4-a437-9847e6da0821#deployment)
9. [Troubleshooting](https://claude.ai/chat/9e5a005f-cc9c-44d4-a437-9847e6da0821#troubleshooting)

## **Wat is dit?**

Een **complete website generator** die automatisch professionele websites maakt met:

* ✅ Moderne HTML/CSS/JavaScript templates
* ✅ Progressive Web App (PWA) functionaliteit
* ✅ SEO optimalisatie (Google rankings)
* ✅ Performance optimalisatie (95+ Lighthouse scores)
* ✅ Security best practices
* ✅ VS Code + Git + Netlify integratie
* ✅ Alle essentiële web bestanden
* ✅ Google Analytics integratie
* ✅ Automated testing & deployment

**Resultaat:** Een production-ready website in minuten in plaats van weken.

## **Quick Start**

### **Vereisten**

* Python 3.8+ geïnstalleerd
* Node.js 18+ geïnstalleerd
* Git geïnstalleerd
* VS Code (aanbevolen)

### **In 5 Stappen:**

# Stap 1: Genereer je website

python complete-website-generator.py --name mijn-website --business "Mijn Bedrijf" --domain mijnbedrijf.nl

# Stap 2: Ga naar project folder

cd projects/mijn-website

# Stap 3: Installeer dependencies

npm install

# Stap 4: Start development server

npm run dev

# Stap 5: Open in browser

# Ga naar: http://localhost:3000

Je website draait nu lokaal! Open code . om te bewerken in VS Code.

## **Volledige Installatie**

### **1. Python Script Uitvoeren**

**Basis gebruik:**

python complete-website-generator.py

**Met alle opties:**

python complete-website-generator.py \

--name mijn-bedrijf \

--business "Mijn Bedrijf BV" \

--domain mijnbedrijf.nl \

--ga G-XXXXXXXXXX

**Beschikbare opties:**

* --name - Project naam (folder naam)
* --business - Bedrijfsnaam (in content)
* --domain - Website domein
* --ga - Google Analytics 4 Measurement ID

### **2. Project Setup**

Na generatie krijg je deze structuur:

mijn-website/

├── src/ # Source code (hier bewerk je)

│ ├── index.html # Hoofdpagina

│ ├── assets/

│ │ ├── css/ # Stylesheets

│ │ └── js/ # JavaScript

│ └── 404.html # Error pagina

├── public/ # Static files (kopiëren naar dist/)

│ ├── robots.txt # Zoekmachine instructies

│ ├── sitemap.xml # Sitemap

│ ├── manifest.json # PWA config

│ └── .htaccess # Server config

├── dist/ # Build output (gegenereerd)

├── package.json # NPM config

├── netlify.toml # Netlify config

└── README.md # Project docs

### **3. Development Environment**

**VS Code Extensions (aanbevolen):**

{

"recommendations": [

"esbenp.prettier-vscode", // Code formatting

"dbaeumer.vscode-eslint", // JavaScript linting

"ritwickdey.liveserver", // Live reload

"eamodio.gitlens", // Git integration

"netlify.netlify-vscode" // Netlify tools

]

}

Installeren: VS Code → Extensions → Install Recommended

## **Project Structuur**

### **Source Files (src/)**

Hier bewerk je content en styling:

src/

├── index.html # Hoofdpagina - bewerk content hier

├── assets/

│ ├── css/

│ │ └── main.css # Styling - pas kleuren/layout aan

│ └── js/

│ └── main.js # Interactiviteit - voeg features toe

├── components/ # Herbruikbare onderdelen

├── pages/ # Extra pagina's (contact, over ons)

└── 404.html # Error pagina

### **Public Files (public/)**

Static bestanden die niet veranderen:

public/

├── robots.txt # Zoekmachine crawling

├── sitemap.xml # URL lijst voor Google

├── manifest.json # PWA configuratie

├── service-worker.js # Offline functionaliteit

├── .htaccess # Apache server config

├── favicon.ico # Browser icon

└── .well-known/

└── security.txt # Security contact

### **Build Output (dist/)**

Geoptimaliseerde versie voor productie (automatisch gegenereerd):

dist/

├── index.html # Minified HTML

├── assets/

│ ├── css/

│ │ └── main.min.css # Gecomprimeerde CSS

│ └── js/

│ └── main.min.js # Gecomprimeerde JavaScript

└── [alle public files] # Gekopieerde static files

## **Alle Bestanden Uitgelegd**

### **Security & Compliance**

**security.txt** (/.well-known/security.txt)

* Contactinfo voor security researchers
* Verplicht voor professionele sites
* RFC 9116 compliant

**GDPR Compliance**

* Cookie consent in HTML
* Privacy policy link
* Data processing transparency

### **SEO Bestanden**

**robots.txt** (/robots.txt)

Wat het doet: Vertelt Google/Bing welke pagina's te indexeren

Waarom belangrijk: Controleert wat in zoekmachines verschijnt

Bewerk voor: Blokkeren van private pagina's

**sitemap.xml** (/sitemap.xml)

Wat het doet: Lijst van alle pagina's voor zoekmachines

Waarom belangrijk: Snellere indexering, betere rankings

Update bij: Nieuwe pagina's toevoegen

**structured-data.json** (in HTML)

Wat het doet: Rich snippets (sterren, prijzen) in Google

Waarom belangrijk: Hogere click-through rates

Schema types: Organization, LocalBusiness, Product, FAQ

### **Performance Bestanden**

**service-worker.js** (/service-worker.js)

Wat het doet: Offline support, caching, snellere laadtijden

Waarom belangrijk: PWA functionaliteit, betere scores

Cache strategie: Network First voor HTML, Cache First voor assets

**manifest.json** (/manifest.json)

Wat het doet: Installeerbare app op mobiel/desktop

Waarom belangrijk: App-achtige ervaring zonder App Store

Configureer: Naam, kleuren, iconen

**critical.css** (inline in HTML)

Wat het doet: Above-the-fold styling voor snelle render

Waarom belangrijk: Betere First Contentful Paint

Optimaliseer: Minimale CSS voor zichtbare content

### **Server Configuratie**

**.htaccess** (Apache/Litespeed)

Features:

- HTTPS redirect (force SSL)

- Clean URLs (zonder .html)

- Security headers

- Browser caching

- Gzip compression

**nginx.conf** (Nginx servers)

Features:

- SSL/TLS configuratie

- Load balancing

- Rate limiting

- Security headers

**web.config** (Windows/IIS)

Features:

- IIS rewrite rules

- MIME types

- Custom errors

### **Development Files**

**package.json**

{

"scripts": {

"dev": "Start development server",

"build": "Build for production",

"preview": "Preview production build",

"deploy": "Deploy to Netlify",

"test": "Run tests",

"lighthouse": "Performance check"

}

}

**.gitignore**

Excludes:

- node\_modules/ (dependencies)

- dist/ (build output)

- .env (secrets)

- .DS\_Store (Mac files)

**netlify.toml**

Configures:

- Build command

- Publish directory

- Redirects

- Headers

- Environment variables

## **Development Workflow**

### **Dagelijkse Workflow**

**1. Start Development Server**

npm run dev

* Website op: http://localhost:3000
* Auto-reload bij wijzigingen
* Hot Module Replacement (HMR)

**2. Maak Wijzigingen**

# Bewerk in VS Code

code src/index.html # Content aanpassen

code src/assets/css/main.css # Styling wijzigen

code src/assets/js/main.js # Functionaliteit toevoegen

**3. Test Lokaal**

* Check in browser
* Responsive design testen (F12 → Device toolbar)
* Console errors checken (F12 → Console)

**4. Commit naar Git**

git add .

git commit -m "feat: beschrijving van wijziging"

git push origin main

**5. Automatische Deploy**

* Netlify detecteert push
* Bouwt automatisch
* Deploy naar productie

### **Git Workflow Best Practices**

**Feature Development:**

# Maak feature branch

git checkout -b feature/nieuwe-functie

# Werk aan feature

# ... edit files ...

# Commit changes

git add .

git commit -m "feat: nieuwe functie toegevoegd"

# Push en maak PR

git push origin feature/nieuwe-functie

**Commit Message Conventies:**

feat: Nieuwe functionaliteit

fix: Bug fix

docs: Documentatie update

style: Code formatting

refactor: Code restructuring

test: Test toevoegingen

chore: Maintenance taken

## **Kwaliteitscontrole & Testing**

### **Lighthouse Checks (Geautomatiseerd)**

**Wat meet Lighthouse:**

1. **Performance (0-100)**
   * First Contentful Paint < 1.8s
   * Largest Contentful Paint < 2.5s
   * Cumulative Layout Shift < 0.1
   * Total Blocking Time < 200ms
2. **Accessibility (0-100)**
   * Alt text op afbeeldingen
   * Kleurcontrast 4.5:1
   * ARIA labels
   * Keyboard navigatie
3. **Best Practices (0-100)**
   * HTTPS gebruik
   * Geen console errors
   * Moderne APIs
   * Veilige dependencies
4. **SEO (0-100)**
   * Meta descriptions
   * Mobile-friendly
   * Crawlable links
   * Valid structured data
5. **PWA**
   * Service worker
   * Offline support
   * Installeerbaar
   * HTTPS

**Lighthouse Uitvoeren:**

**In Browser:**

1. Open website in Chrome

2. F12 (DevTools)

3. Tab "Lighthouse"

4. Click "Analyze page load"

5. Bekijk scores en tips

**Via Command Line:**

npm run lighthouse

**In CI/CD (Automatisch):**

* Draait bij elke deploy
* Faalt bij scores < 90%
* Rapport in Netlify dashboard

### **Extra Testing Tools**

**Performance:**

# Google PageSpeed Insights

https://pagespeed.web.dev/

# WebPageTest

https://www.webpagetest.org/

# GTmetrix

https://gtmetrix.com/

**Security:**

# Security Headers Check

https://securityheaders.com/

# SSL Test

https://www.ssllabs.com/ssltest/

# Mozilla Observatory

https://observatory.mozilla.org/

**SEO:**

# Google Search Console

https://search.google.com/search-console

# Schema Validator

https://validator.schema.org/

# Mobile-Friendly Test

https://search.google.com/test/mobile-friendly

### **Kwaliteits Checklist**

**Pre-Launch:**

□ Lighthouse scores >90 alle categorieën

□ Geen console errors

□ Cross-browser getest (Chrome, Safari, Firefox)

□ Responsive op mobiel/tablet/desktop

□ Alle links werken

□ Formulieren getest

□ Analytics tracking werkt

□ Favicon zichtbaar

□ 404 pagina werkt

□ HTTPS actief

**Post-Launch:**

□ Google Search Console ingesteld

□ Google Analytics data komt binnen

□ Sitemap gesubmit

□ Uptime monitoring actief

□ Backup systeem werkt

□ Error tracking configured

## **Deployment**

### **Netlify Setup (Aanbevolen)**

**Eerste Deployment:**

1. **Account Aanmaken**
   * Ga naar netlify.com
   * Sign up met GitHub

**Site Connecten** # In je project folder

netlify login

netlify init

1. **Build Settings Configureren**
   * Build command: npm run build
   * Publish directory: dist
   * Production branch: main

**Deploy** npm run deploy

**Automatische Deploys:**

* Push naar main → Auto deploy productie
* Pull Request → Preview deployment
* Rollback met 1 click mogelijk

**Custom Domain:**

# In Netlify dashboard

1. Domain settings

2. Add custom domain

3. Update DNS bij provider:

- A record: 75.2.60.5

- CNAME: [site-name].netlify.app

### **Alternatieve Hosting**

**Vercel:**

npm i -g vercel

vercel --prod

**GitHub Pages:**

# .github/workflows/deploy.yml

name: Deploy

on: [push]

jobs:

build-deploy:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- run: npm ci

- run: npm run build

- uses: peaceiris/actions-gh-pages@v3

**Traditional Hosting (cPanel/FTP):**

1. npm run build

2. Upload dist/ folder via FTP

3. Zet document root naar dist/

## **Troubleshooting**

### **Veelvoorkomende Problemen**

**"npm: command not found"**

# Installeer Node.js

# Download van: nodejs.org

# Herstart terminal

node --version # Check installatie

**"Permission denied" bij npm install**

# Mac/Linux

sudo npm install

# Of fix npm permissions:

mkdir ~/.npm-global

npm config set prefix '~/.npm-global'

echo 'export PATH=~/.npm-global/bin:$PATH' >> ~/.profile

**Build faalt op Netlify**

# Check build logs

# Veelvoorkomende fixes:

1. Netlify.toml Node version updaten

2. package-lock.json committen

3. Dependencies checken met: npm audit fix

**Lighthouse lage scores**

# Performance:

- Optimaliseer afbeeldingen (WebP, lazy loading)

- Minimaliseer JavaScript/CSS

- Gebruik CDN voor libraries

# Accessibility:

- Alt text toevoegen

- Contrast verhogen

- ARIA labels toevoegen

# SEO:

- Meta descriptions schrijven

- Structured data valideren

- Mobile-friendly maken

**Service Worker werkt niet**

# Check:

1. HTTPS actief? (verplicht)

2. service-worker.js in public/?

3. Browser cache legen

4. Incognito mode testen

### **Debug Tips**

**Development:**

# Verbose logging

npm run dev -- --debug

# Check build output

npm run build -- --verbose

# Test production locally

npm run preview

**Browser DevTools:**

F12 → Console: JavaScript errors

F12 → Network: Laadtijden & failed requests

F12 → Application: Service Worker, Storage

F12 → Lighthouse: Performance audit

## **Support & Resources**

### **Documentatie**

* [MDN Web Docs](https://developer.mozilla.org/) - Web technologie reference
* [web.dev](https://web.dev/) - Google best practices
* [Netlify Docs](https://docs.netlify.com/) - Deployment guide

### **Community**

* Stack Overflow - Technische vragen
* GitHub Issues - Bug reports
* Discord/Slack - Real-time hulp

### **Handige Tools**

* [Can I Use](https://caniuse.com/) - Browser compatibility
* [TinyPNG](https://tinypng.com/) - Image compression
* [Google Fonts](https://fonts.google.com/) - Web fonts
* [Heroicons](https://heroicons.com/) - SVG icons
* [Coolors](https://coolors.co/) - Color schemes

## **Licentie & Credits**

**MIT License** - Vrij te gebruiken voor commerciële projecten

**Gebouwd met:**

* Python 3.x
* Vite - Modern build tool
* Netlify - Hosting & deployment
* Google Analytics - User tracking
* Lighthouse - Quality assurance

**Gegenereerd met:** Ultra Professional Website Generator v1.0

*Voor vragen of problemen, maak een issue aan op GitHub of neem contact op via email.*

**Happy coding! 🚀**